Lab Exercise 4– Terraform Variables

Objective:

Learn how to define and use variables in Terraform configuration.

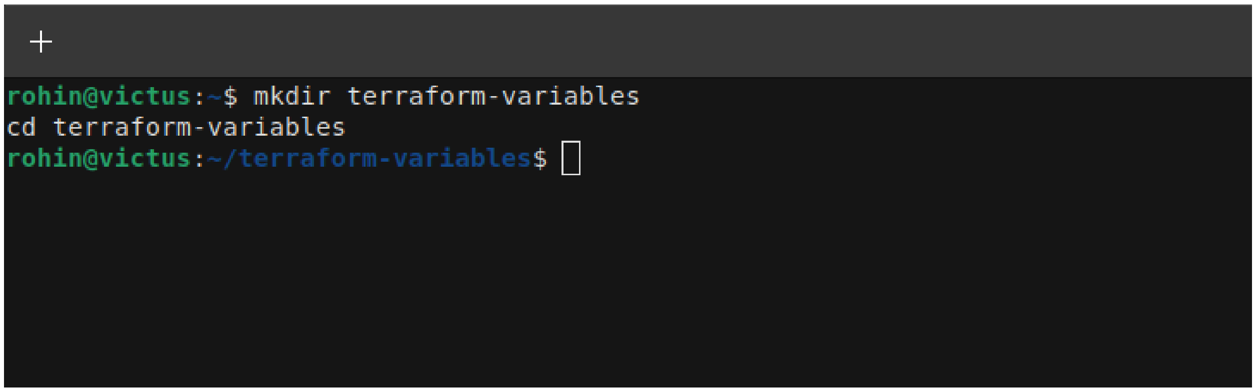
Prerequisites:

•Install Terraform on your machine.

Steps:

1. Create a Terraform Directory:

•Create a new directory for your Terraform project.

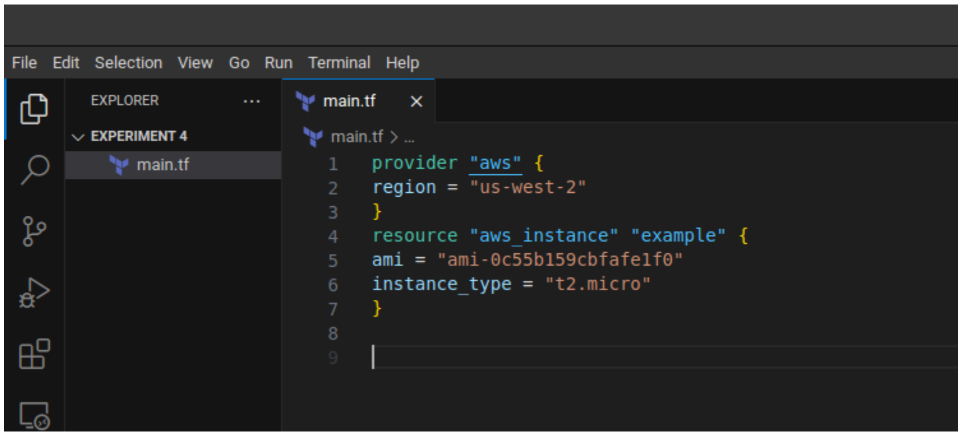


```
rohin@victus:~$ mkdir terraform-variables
cd terraform-variables
rohin@victus:~/terraform-variables$
```

2. Create a Terraform Configuration File:

•Create a file named main.tf within your project directory.

main.tf

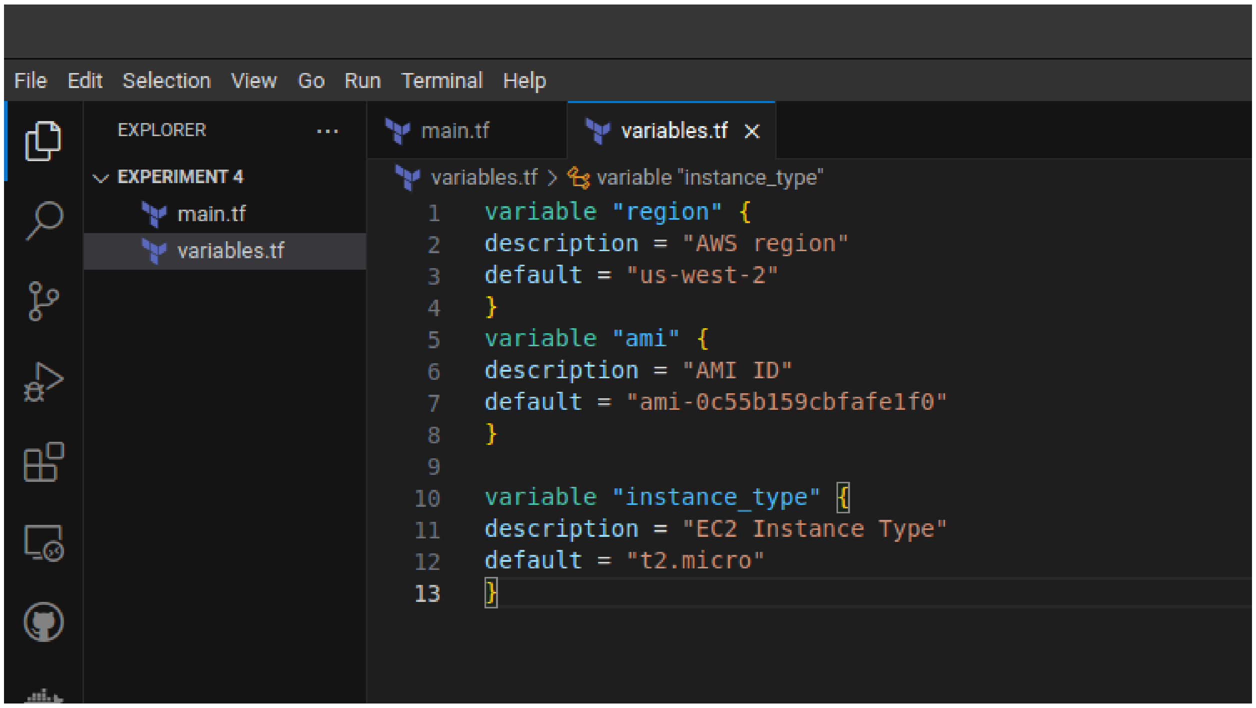


```
provider "aws" {
region = "us-west-2"
}
resource "aws_instance" "example" {
ami = "ami-0c55b159cbfafe1f0"
instance_type = "t2.micro"
}
```

3. Define Variables:

•Open a new file named variables.tf. Define variables for region, ami, and Instance_type.

# variables.tf


File Edit Selection View Go Run Terminal Help
EXPLORER
EXPERIMENT 4
main.tf
variables.tf
variables.tf > variable "instance_type"

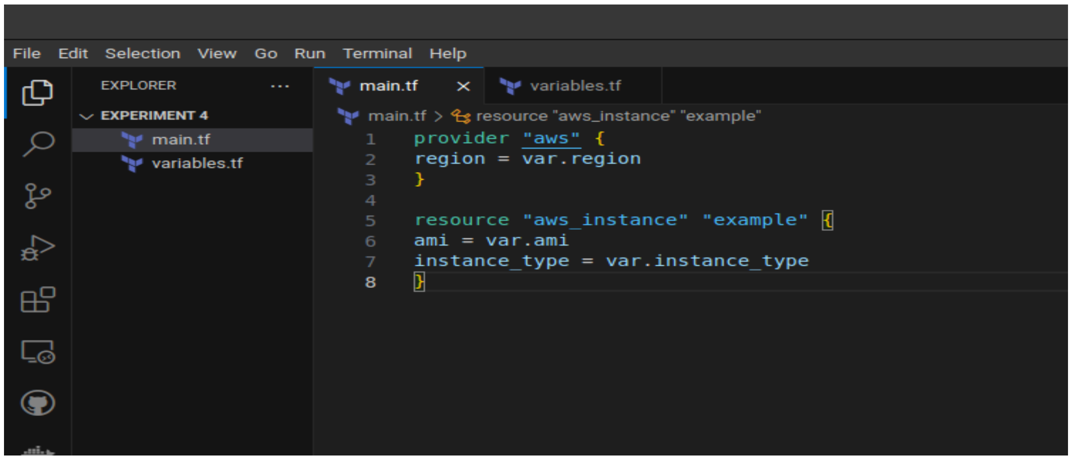
```
variable "region" {
description = "AWS region"
default = "us-west-2"
}
variable "ami" {
description = "AMI ID"
default = "ami-0c55b159cbfafe1f0"
}

variable "instance_type" {
description = "EC2 Instance Type"
default = "t2.micro"
}
```

4. Use Variables in main.tf:

•Modify main.tf to use the variables.

# main.tf


File Edit Selection View Go Run Terminal Help
EXPLORER
EXPERIMENT 4
main.tf
variables.tf
main.tf > resource "aws_instance" "example"

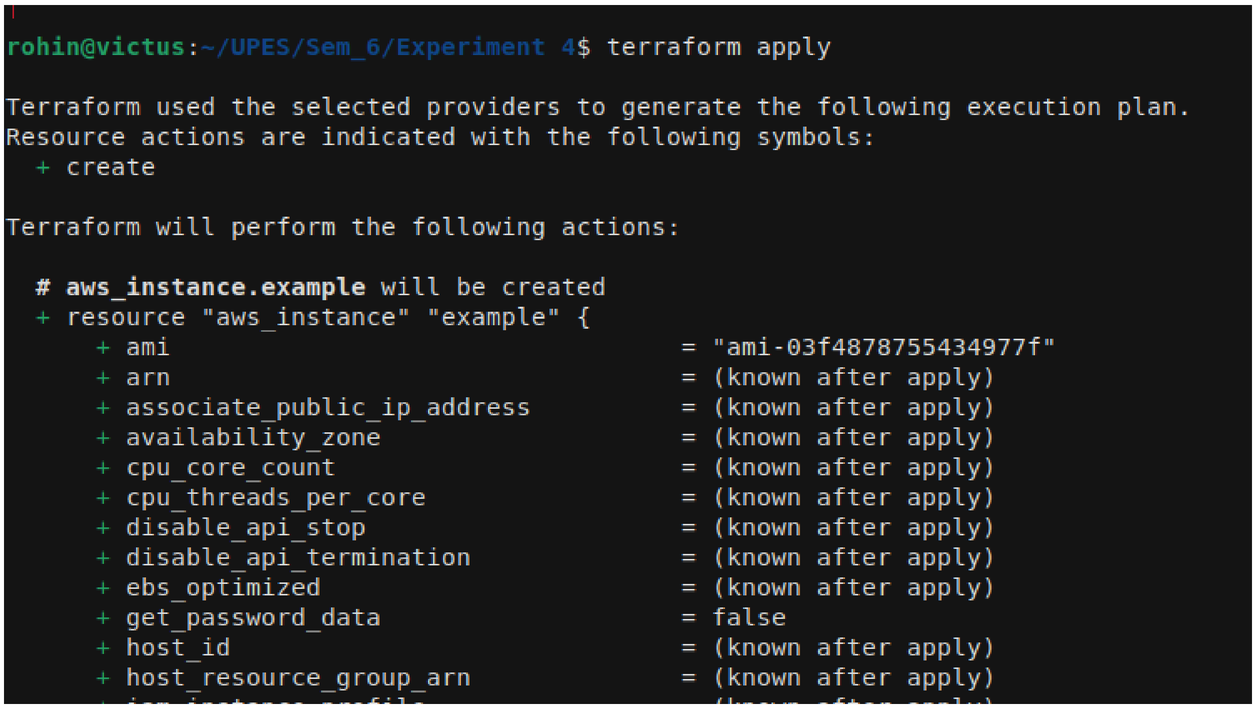
```
provider "aws" {
region = var.region
}

resource "aws_instance" "example" {
ami = var.ami
instance_type = var.instance_type
}
```

\\

5. Initialize and Apply:

•Run the following Terraform commands to initialize and apply the configuration.



```
rohin@victus:~/UPES/Sem_6/Experiment 4$ terraform apply

Terraform used the selected providers to generate the following execution plan.
Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.example will be created
  + resource "aws_instance" "example" {
      + ami                                  = "ami-03f4878755434977f"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = (known after apply)
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
```

Observe how the region changes based on the variable override.

6. Clean Up:

After testing, you can clean up resources.

`terraform destroy`
Confirm the destruction by typing yes.

7. Conclusion:

This lab exercise introduces you to Terraform variables and demonstrates how to use them in your configurations. Experiment with different variable values and overrides to understand their impact on the infrastructure provisioning process.